HOME AUTOMATION PROJECT

(Smart-Fox)

A PROJECT REPORT

SUBMITTED IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE AWARD OF THE DEGREE OF

BACHELOR OF TECHNOLOGY

In

COMPUTER SCIENCE ENGINEERING

By

Rajan Jiten Patel
Reg. No. 149105228

MANIPAL UNIVERSITY
JAIPUR

INSPIRED BY LIFE

DEPARTMENT OF COMPUTER SCIENCE ENGINEERING

MANIPAL UNIVERSITY JAIPUR
JAIPUR 303 007, RAJASTHAN, INDIA

JULY - 2018

DEPARTMENT OF COMPUTER SCIENCE

MANIPAL UNIVERSITY JAIPUR, JAIPUR – 303 007 (RAJASTHAN), INDIA

Date: 10th July 2018

CERTIFICATE

This is to certify that the **home automation project** is a record of the bonafide work done by **Rajan Jiten Patel** (149105228) submitted in partial fulfilment of the requirement for the award of the Degree of Bachelor of Technology (B.Tech) in Computer Science of Manipal University Jaipur, during the academic year 2017-18.

Mr. Mahesh Jangid,

Project Guide, Dept of Computer Science

Manipal University Jaipur

Dr Satish Chandra Kulhari,

HOD, Dept of Computer Science

Manipal University Jaipur.

ACKNOWLEDGMENTS

I take this opportunity to express my profound gratitude and deep regards to my guide Mr. Paras Shah, for his exemplary guidance, monitoring and constant encouragement throughout the course of this project. The help and guidance given by him time to time shall carry me a long way in the journey of life on which I am about to embark.

I also take this opportunity to express a deep sense of gratitude to Mr. Mahesh Jangid, for his cordial support, valuable information and guidance, which helped me in completing this task through various stages.

I am obliged to staff members of Softweb Solution, for the valuable information provided by them in their respective fields. I am grateful for their cooperation during the period of my project internship.

Lastly, I thank all the Department of Computer Science faculty for their constant encouragement without which this project would not be possible.

ABSTRACT

We live in an exciting time where more and more everyday items “things” are becoming smart! “Things” have sensors and can communicate with other “things” and can provide control to more “things”. The Internet of Things, IoT, is upon us in a huge way and people are rapidly inventing new gadgets that enhance our lives. The price of microcontrollers with the ability to talk over a network keeps dropping and developers can now think and build things inexpensively.

This IoT based home automation project is done using low-cost custom build ESP8266 WiFi Module, and Raspberry pi zero. A system that uses mobile app to control basic home functions and features automatically through internet from anywhere around the world, an automated home is sometimes called a smart home. It is meant to save the electric power and provide better security service. The home automation system differs from other systems by allowing the user to operate the system from anywhere around the world through internet connection without breaking any convention usability.

This project provides significant electric power saving solution at homes and at the offices with help of dynamic control of electric supply over internet using smart switch, and another is a low-cost wireless plug and use surveillance system with live streaming over internet connection and in-build video storage capacity.

Project contains both hardware and software development where hardware program is built using C and python language on Arduino IDE, whereas mobile application is built using java language Android studio.

LIST OF TABLE

LIST OF FIGURES

Chapter 1

INTRODUCTION

1.1 Introduction

Home automation has made it possible to have what is often referred to as a "Smart Home", a home that can detect and identify you, automatically adjust the lighting to your predefined taste, open doors automatically, play your favorite music, water your flower in the morning, switch on the security lights at night and switch them off in the morning, heat water for bathe and tea, stream to you anywhere in the world via internet a live video of what is happening in and around your premises. It make it possible to link lighting, entertainment, security, telecommunications, heating, and air conditioning into one centrally controlled system. This allows you to your house an active partner in managing your busy life and assist you with efficient living.

Nowadays, you can hardly find a house without a home automation system which can range from the remote for the television, burglar alarm and hi-tech security surveillance, to an automated air condition system that maintain the temperature at a predefined value.

This project wants to add new devices to home which can progress towards fully automated home and make life easier and productive.

1.2 Objectives of the Project

The objective of this project is to design and build new IoT hardware which can be control using mobile application from anywhere around the world and that can installed at every home and office at lowest possible cost.

1.3 Project Focus

Project has focus on two main area of home which are electricity and security, and come up with two hardware solution:-

I. Smart Switch.
II. IP-based Surveillance Cam.

1.4 Organization of Report

The entire Project is composed of five chapters. each covering a section of work as summarized below:

- Chapter one gives an introduction to automation as a whole and the different types of automation. And also define project motivation and objectives.
- Chapter two covers an extensive background overview of project working and which technology are involved in development with establishment of standards and protocols and benefits.
- Chapter three highlights the project methodology, giving reasons for choice of specific platform and components, and also comprehensive details on project development timeline.
- Chapter four is on the project design and implementation with clear practical details of the project design, construction, testing, microcontroller coding and debugging. Special emphasis is also made on the flexibility and scalability of the project work with real life illustration.
- Chapter five is on the project results and analysis, that how project fill up the needs which was analysis at initial stage.
- Last chapter is on the conclusion and recommendations based on the project work with emphasis on the reliability, maintainability and flexibility of the design. Also, recommendations based on the challenges encountered and further possible development of the project work.

Chapter 2

BACKGROUND MATERIAL

2.1 Conceptual Overview

Project developed two devices:-

- Smart-Switch
- IP-based Surveillance Camera

2.1.1 Conceptual view of Smart-Switch

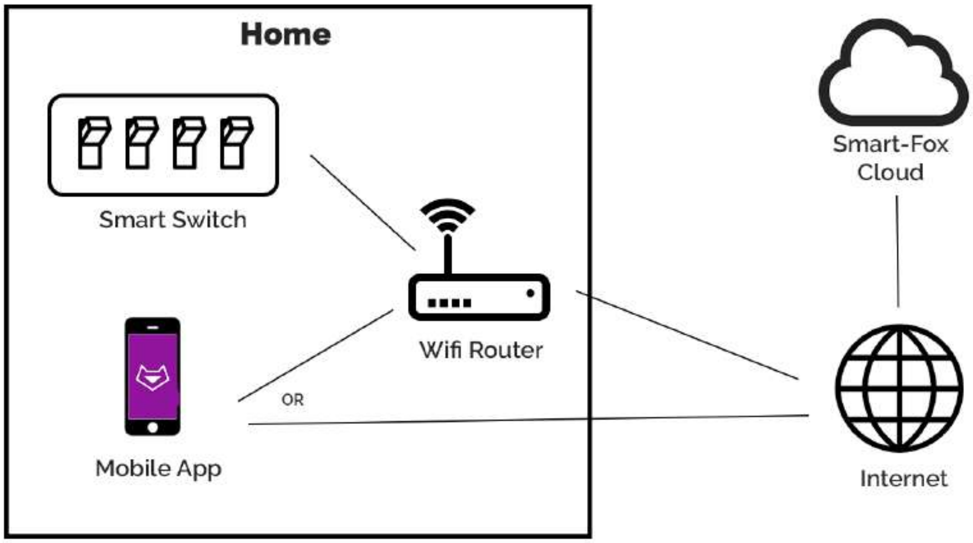


Fig 2.1 Smart-Switch overview

- Smart switch is operated by ESP8266 with Wi-Fi-module hardware which control and connect switch to internet though local Wi-Fi network.
- Mobile app can connect to smart-switch using local Wi-Fi network and also using GSM internet or other Wi-Fi network.
- Cloud provide accessible to control switch though internet.
- Smart-Switch can be operated using manual switch and also through mobile app. Live status of switch can be monitor from mobile app even if switch get used using manually.

2.1.2 Conceptual view of IP-based Surveillance Camera

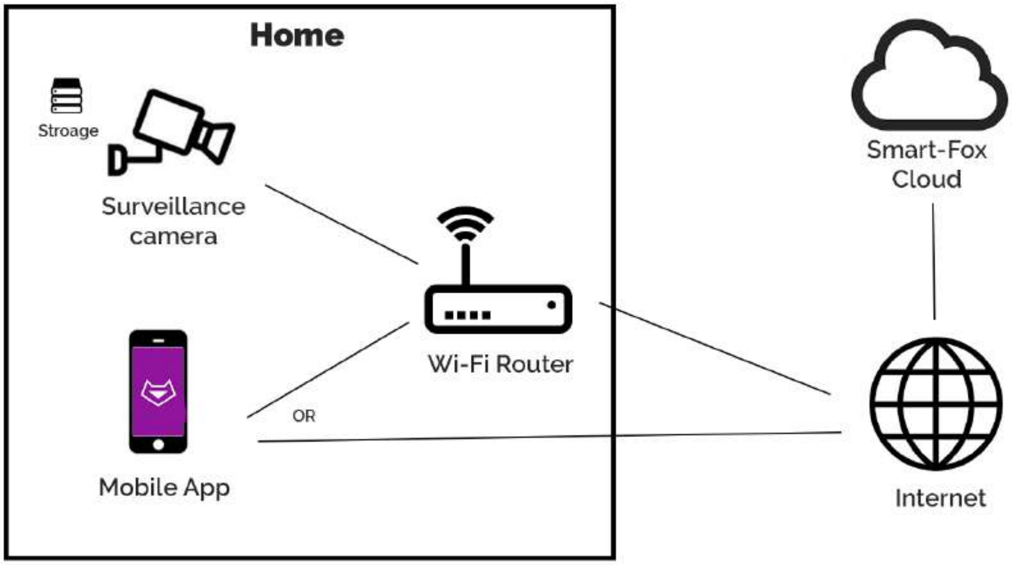


Fig 2.2 IP-based surveillance cam.

- Surveillance camera is operated on raspberry pi zero hardware and serially connect with Sony IMX219 image sensor with night vision capacity which allow to capture 1080p motion sense video.
- Surveillance camera has its on video storage on-board can record up to 24 hours video with re-writable over a loop and can also add long storage capacity by using cloud.
- Live video can be stream using mobile app through internet.
- Surveillance camera can be used as plug and use, module required only power supply though micro-usb, so it can be moved anywhere as per need within Wi-Fi range.

2.2 Technologies Involved

2.2.1 Technology using for Smart-Switch

Hardware used to build Smart-Switch.

- NodeMCU ESP8266 Wi-Fi module.

  Type - 32-bit micro-controller
  CPU - 160MHz
  GPIO -16 pin
  Memory - 4MB

- 4 two-way switches.
- 4 port electromagnet relay.
- 220 Volt to 5 Volt AC to DC power adapter.

Software used to program ESP8266 microcontroller

- Arduino IDE using C language.

Development of mobile application.

- Android Studio 3.1.1 - IntelliJ platform.

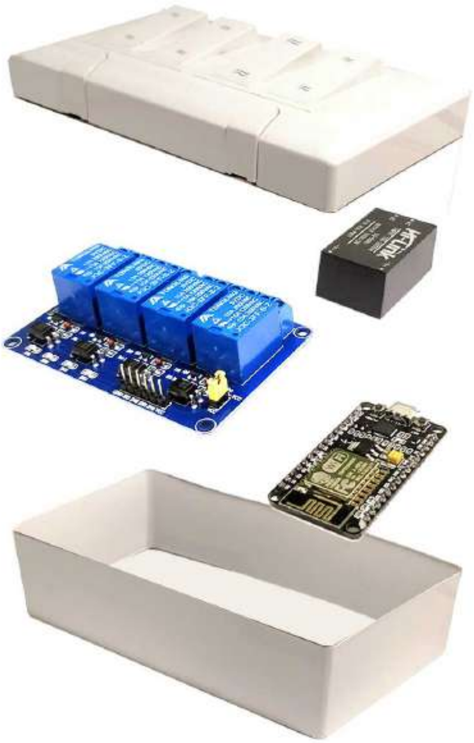


Fig 2.3 Switch Setup

NodeMCU ESP8266 Wi-Fi module [1]

The NodeMCU (Node MicroController Unit) is an open source software and hardware development environment that is built around a very inexpensive System-on-a-Chip (SoC) called the ESP8266. The ESP8266, designed and manufactured by Espressif Systems, contains all crucial elements of the modern computer: CPU, RAM, networking (Wi-Fi), and even a modern operating system and SDK. When purchased at bulk, the ESP8266 chip costs only $2 USD a piece. That makes it an excellent choice for this project.

Fig 2.4 NodeMCU

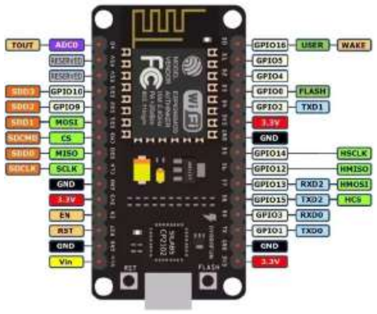


Fig 2.5 NodeMCU architecture diagram

Android Mobile Application

About half of the population around the world prefers Android over another operating systems or devices. IoT is one of the biggest areas where Android app development technology is continuously contributing with its enormous benefits. Internet of Things is generally recognized as the interconnectedness of different smart devices over the Internet. The devices make use of sensors and internet connectivity, which helps them receive, collect and transmit information. So development on android platform is the right option for this project.

Smart-Switch application named Smart-Fox is developed on android platform with API 19 compatible which covers 90.1% android device of the total devices.

List of support used on android device to run mobile application.

- Wi-Fi.
- Internet GSM.
- Location.
- Room SQLite database.

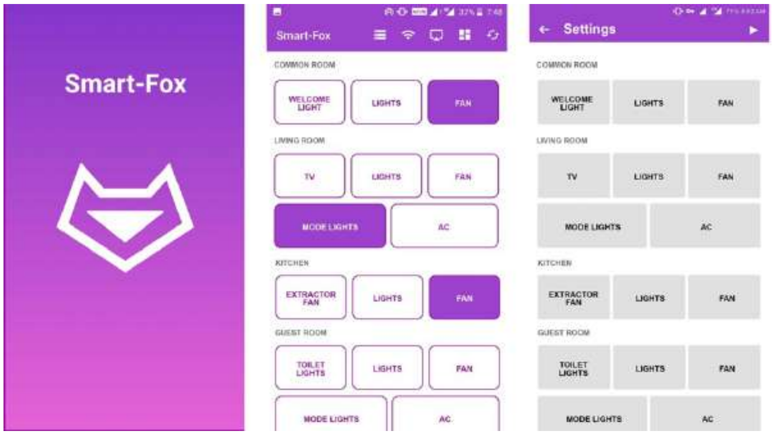


Fig 2.6 Android Mobile Application

2.2.1 Technology using for IP-Based Surveillance Cam

Hardware used to build Surveillance Camera

- Raspberry Pi Zero W.

  - 1GHz, single-core CPU
  - 512MB RAM
  - Mini HDMI and USB On-The-Go ports
  - Micro USB power
  - HAT-compatible 40-pin header
  - Composite video and reset headers
  - CSI camera connector

- Sony IMX219 image sensor.

Software used to build Surveillance Cam

- 2.7 version python platform.
- OpenCV.

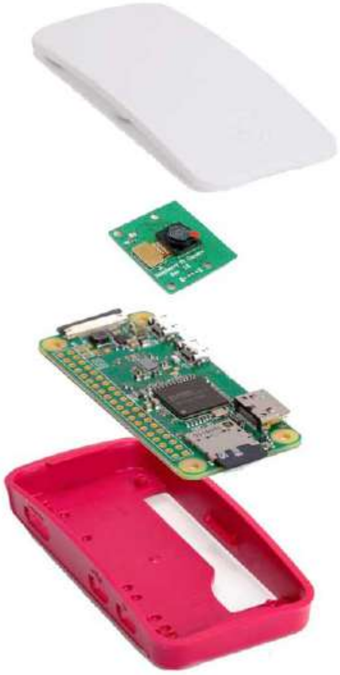


Fig 2.7 Cam Setup

Raspberry Pi Zero W [2]

The Raspberry Pi is a popular Single Board Computer (SBC) in that it is a full computer packed into a single board. Many may already familiar with the Raspberry Pi 3 and its predecessors, which comes in a form factor that has become as highly recognizable. The Raspberry Pi comes in an even smaller form factor. The introduction of the Raspberry Pi Zero allowed one to embed an entire computer in even smaller projects. This project will use the latest version of the Zero product line, the Raspberry Pi Zero - Wireless, which has an onboard Wi-Fi module.

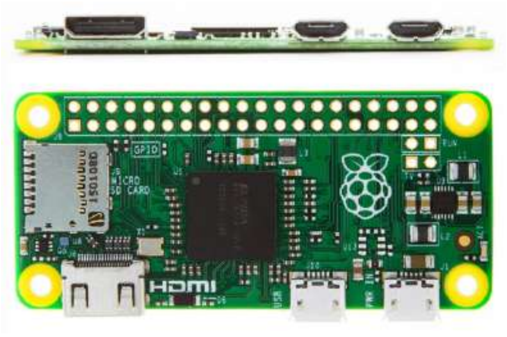


Fig 2.8 Raspberry Pi Zero

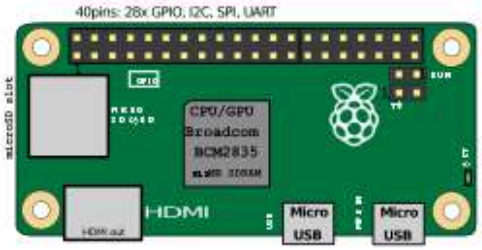


Fig 2.9 Raspberry Pi Zero Architecture

Sony IMX219 image sensor [3]

"IMX219", Type 1/4.0, approximately 8.08M effective pixels, back-illuminated CMOS image sensor for the growing mobile market. The IMX219 satisfies the fulfillment performance of "high sensitivity" and "high frame rate imaging" to meet the demands for high-quality camera applications, and can be easily put on camera fronts of stylish and slim bezel devices achieved by the reduced size.

Not only the normal rate of all-8M pixels at full field of view and 30 frame/s, 4 times faster imaging is also possible by 2 × 2 analog binning mode. Also, a LSC function*1 corrects optical unevenness to adjust during the module fabrication process, and contributes to suppressing of system cost. Additionally, combining with a rear-end ISP supporting of BME-HDR*2 enables future scalability of the high-dynamic-range video imaging.

Image sensor is prefect for video surveillance with low power consumption and high quality video. This make image sensor perfect fit for this project.

Fig 2.10 Image Sensor

2.1.2 Programming Language used during project

Project has used three different programming languages for our project. For the development of the application on android, we have used Java Platform. Android Software Development kit incorporates IntelliJ software where Java programming is performed. IntelliJ software is used to write the codes for the application under Java Platform. Raspbian OS is used at the raspberry pi.

Java: Java is a set of several computer software products and specifications from Oracle Corporation that provides a system for developing application software and deploying it in a cross-platform computing environment. Java is used in a wide variety of computing platforms from embedded devices and mobile phones on the low end, to enterprise servers and supercomputers on the high end.

Python: Python is an interpreter, interactive, object-oriented programming language. It incorporates modules, exceptions, dynamic typing, very high level dynamic data types, and classes. Python combines remarkable power with very clear syntax. It has interfaces too many system calls and libraries, as well as to various window systems, and is extensible in C or C++. It is also usable as an extension language for applications that need a programmable interface. Python is a high-level general-purpose programming language that can be applied to many different classes of problems. 5.2.3.

C: For embedded programming of NodeMCU for Smart Switch using Arduino IDE. it is a non-object-oriented high-level programming language.

Raspbian OS: Raspbian is a free Operating System based on Debian optimized for the raspberry pi hardware. Raspbian comes with more than 35000 packages; pre-combined software bundled in a nice format for easy installation on Raspberry pi.

Chapter 3

METHODOLOGY

3.1 Detailed methodology that will be adopted

This section represent which development cycle project used and how it is implemented.

3.1.1 Software development cycle.

This project used iterative software development[4]. An iteration incorporates a loosely sequential set of activities in business modeling, requirements, analysis and design, implementation, test, and deployment, in various proportions depending on where in the development cycle the iteration is located. Iterations in the inception and elaboration phases focus on management, requirements, and design activities; iterations in the construction phase focus on design, implementation, and test; and iterations in the transition phase focus on test and deployment.

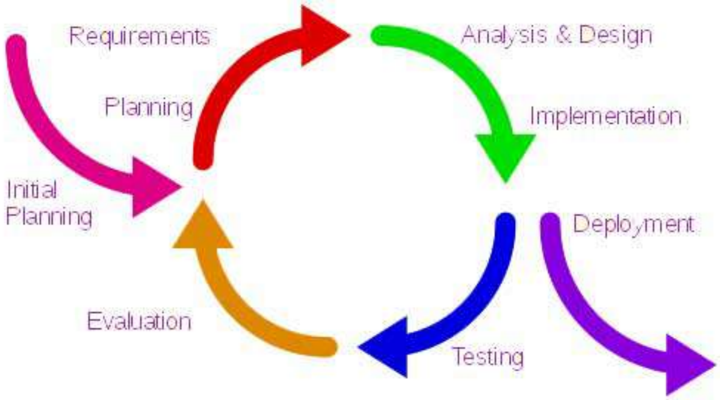


Fig 3.1 Software Development Cycle

Project benefits using Iterative implementation

An iterative approach is generally superior to a linear or waterfall approach for many different reasons.

- Risks are mitigated earlier, because elements are integrated progressively.
- Changing requirements and tactics are accommodated.
- Improving and refining the product is facilitated, resulting in a more robust product.
- Organizations can learn from this approach and improve their process.
- Reusability is increased.

3.2 Overall Project Timeline

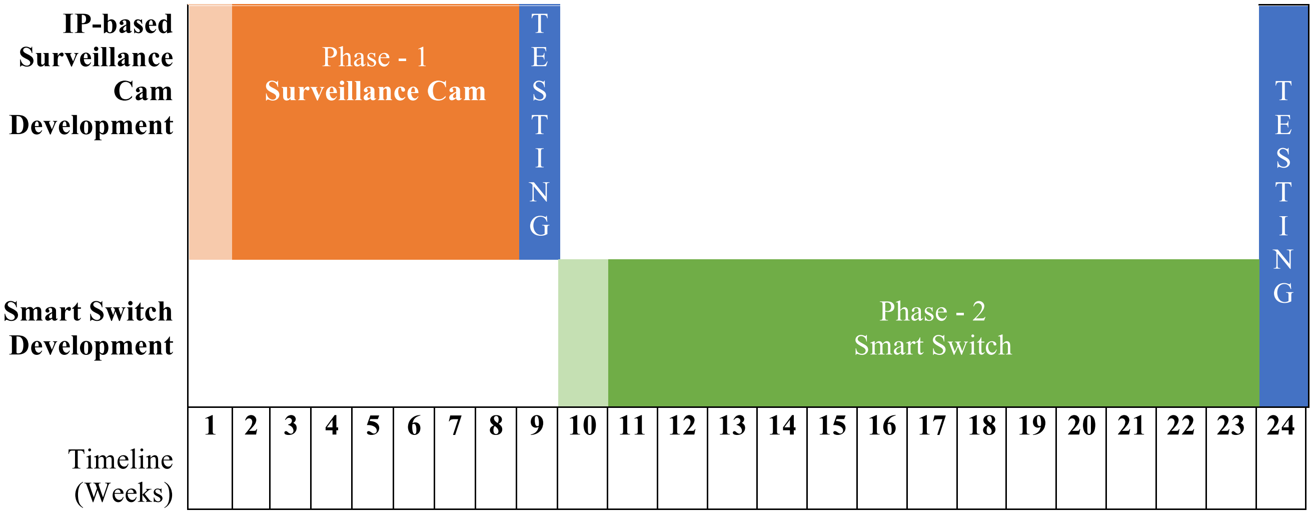


Table 3.1 Overall Timeline

Phase -1 - Development of IP-based Surveillance Cam.

This development was choose at first because it required low amount of time for research to analysis and select perfect hardware which is highly efficient and robust in design as it was used as plug and play. Total three Iterative cycles were required to refine the device in to perfect working condition.

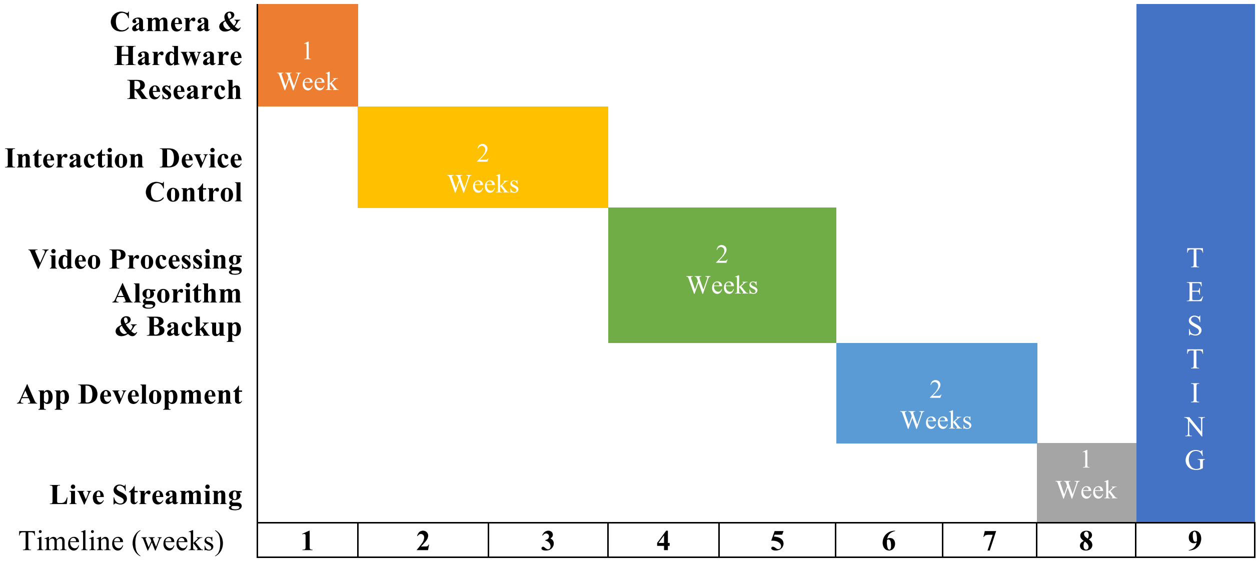


Table 3.2 IP-based Surveillance Cam Timeline

Phase -2 - Development of Smart-Switch.

This development was choose at second because it required high amount of time to develop due to two different platform. Total six Iterative cycles were required to refine the device in to perfect working condition.

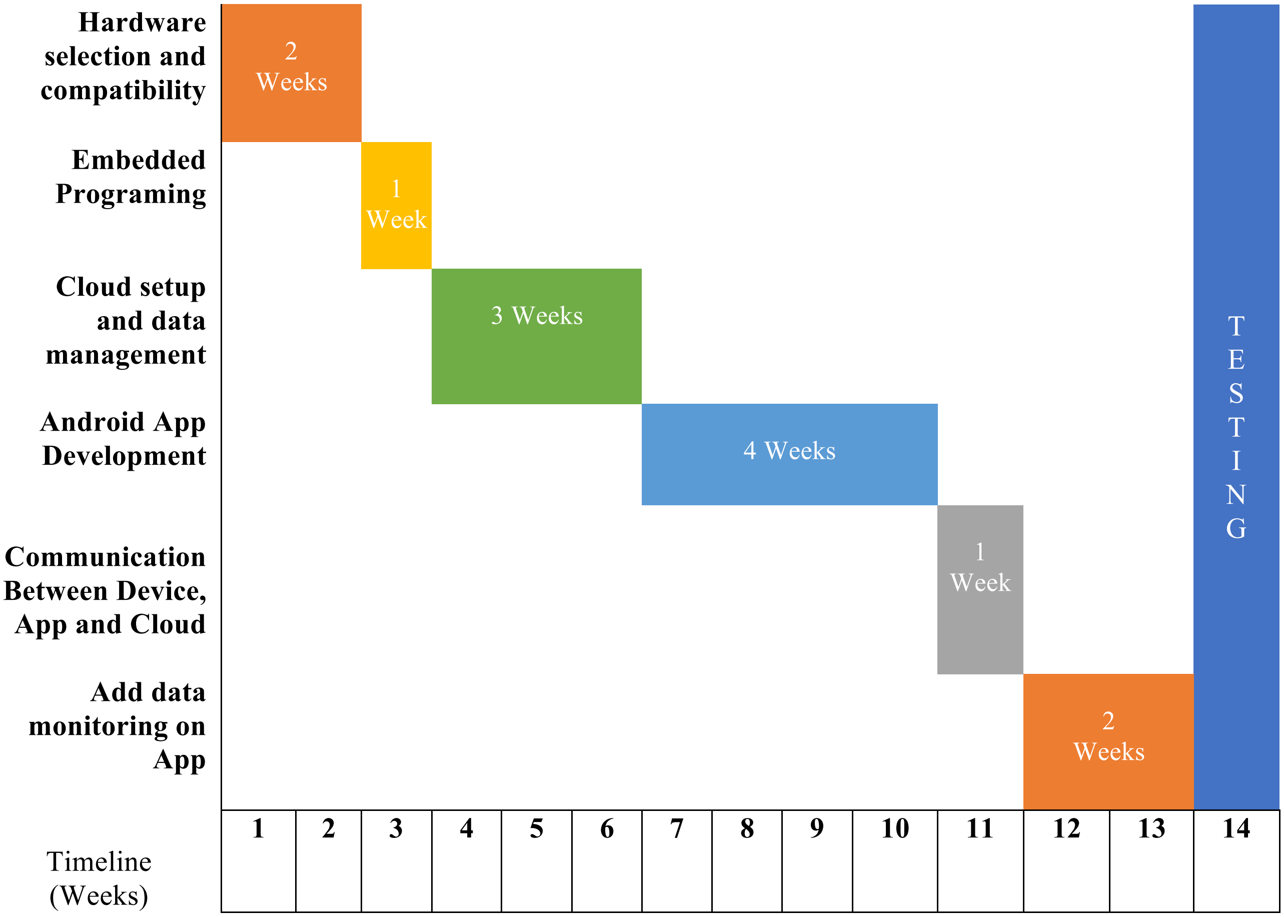


Table 3.3 Smart-Switch Timeline

Chapter 4

IMPLEMENTATION

4.1 Modules

Project is divided into two development modules :-

1) Smart Switch development

1. Hardware circuit development
2. Different possible test case.
3. Embedad sudo program.
4. Web sevice format and data storage format.
5. Mobile application flow and development.

2) IP-based Surveillance Camera development.

1. Hardware circuit development
2. Different test parameters.
3. Embedad sudo program.
4. Web sevice format and data storage format.
5. Mobile application flow and development

4.1 Smart Switch development

1) Hardware Circuit development

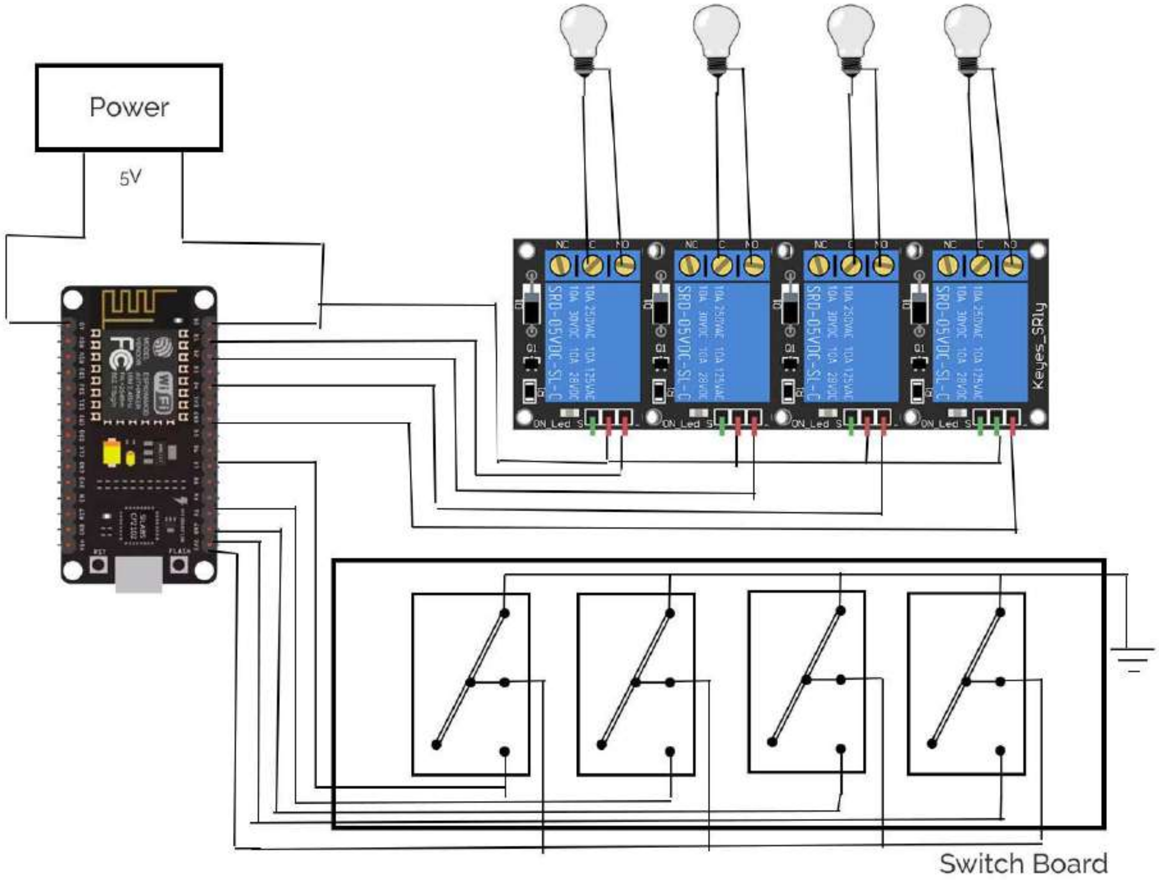


Fig 4.1 - Switch circuit diagram

Except bulb connect, all device and circuit with be packed in switch board. Circuit diagram represent accurate layout of connection inside switch board. Where individual switch are connected with NodeMCU board and operate at 5 volt and electromagnetic trigger relay is also control by NodeMCU board. When switch is on board trigger relay to complete 240 volt circuit of bulb, result to light in bulb.

So in Smart-Switch, NodeMCU board take decision when to turn on light, when switch is pressed manually or receive request from mobile app through Wi-Fi network over internet.

2) Different possible case studies.

Initial case - Smart-switch is connected to web server using local Wi-Fi network over internet. And mobile application can be connected to switch and set server through local Wi-Fi network or over GSM internet connect as shown in Fig 4.2.

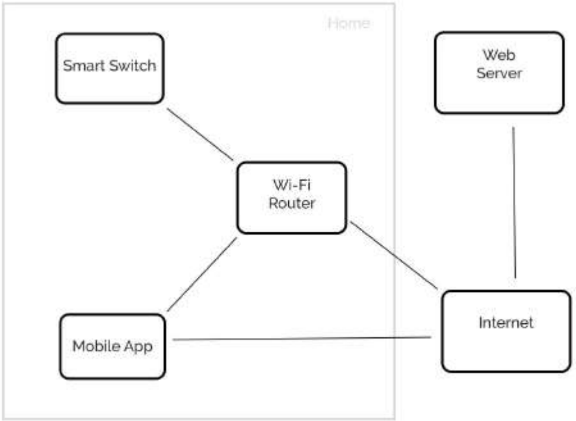


Fig 4.2 Initial Stage

1st Case - when switch is turn on manually. Switch send request to server and update the status, web server also update status of mobile app by sending back request to mobile application. same process also repeat for turn off request.

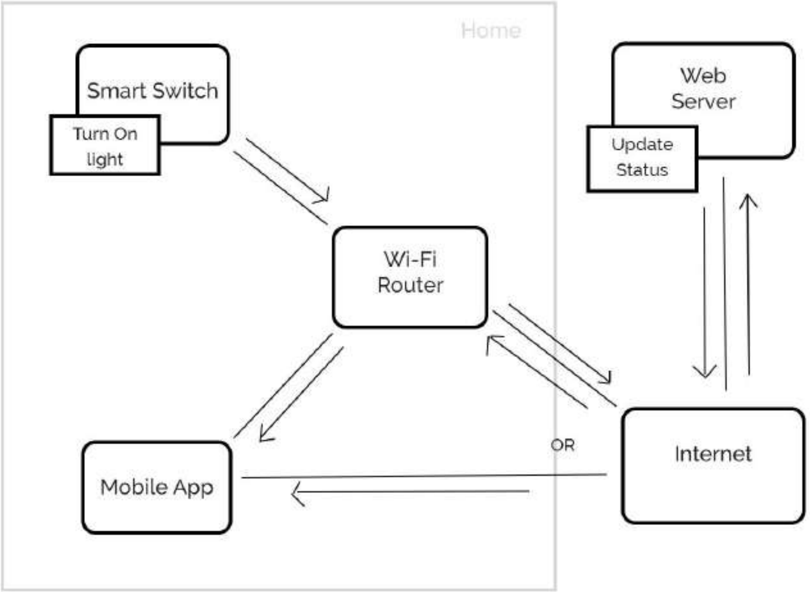


Fig 4.3 First case of Smart switch

2nd Case - when switch is turn on from mobile app when switch is in same Wi-Fi network. Mobile app send direct request to smart switch and update status, same request is also sent to server.

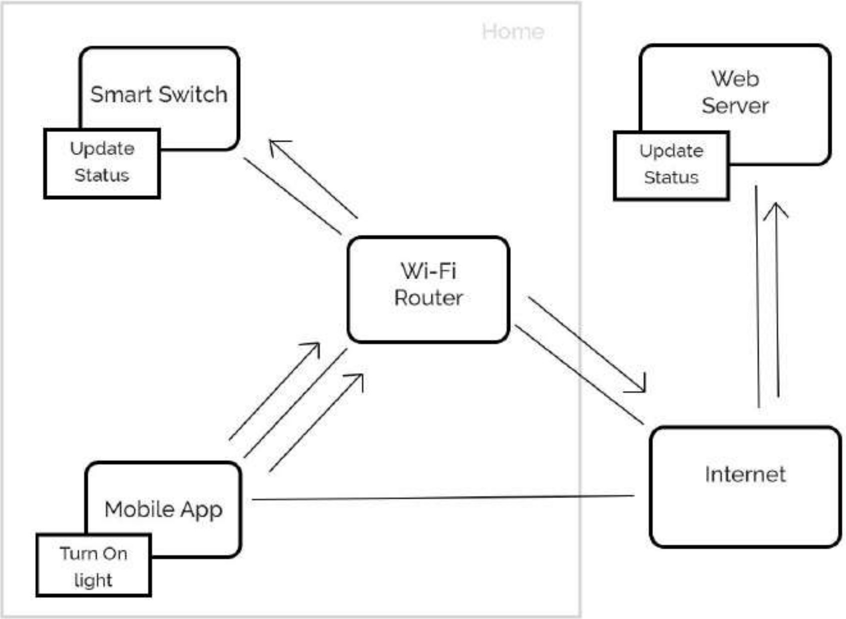


Fig 4.4 Second case Smart Switch.

3rd Case - when switch is turn on from mobile app when connect over internet. Mobile app sent request to web server and Switch get update from web server.

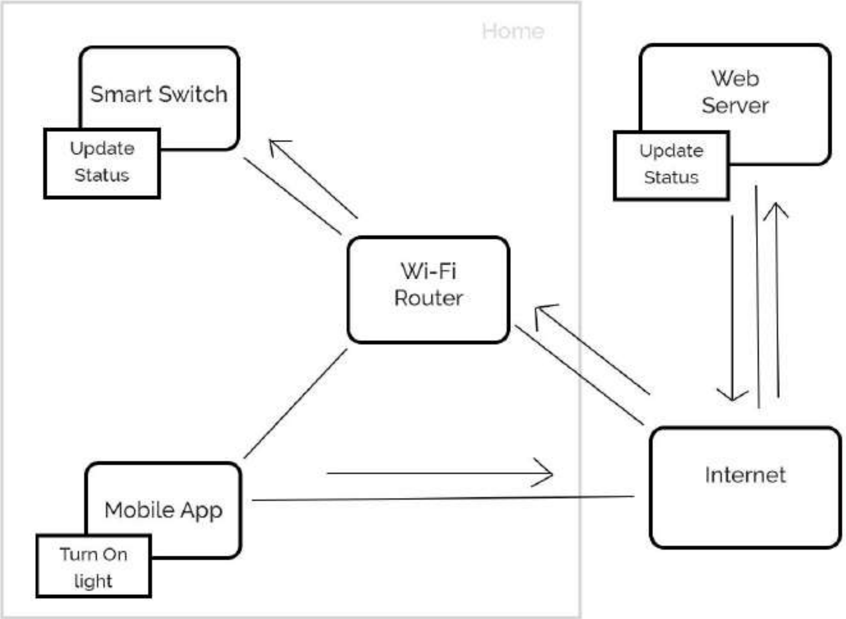


Fig 4.5 Forth case Smart Switch

4th Case - When Switch is turn on from mobile app when to internet services off-line and mobile is connected to same network.

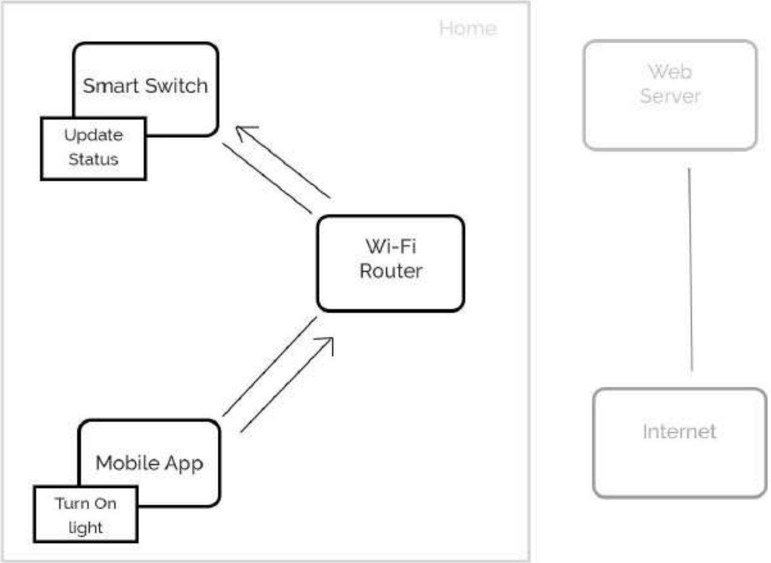


Fig 4.6 Forth case Smart-Switch

Last Case - When smart switch is off line. Even then switch can be turn on and off manually.

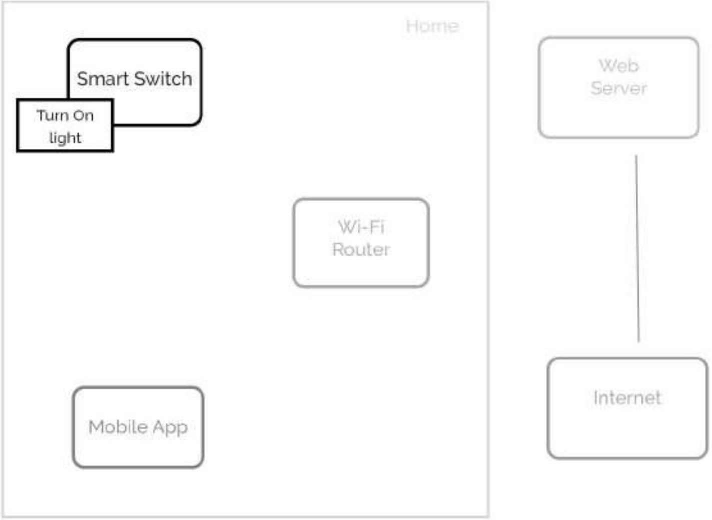


Fig 4.7 Last case smart-switch

3) Embedad sudo program.

```
void start()
{
        get wifi credentials ()
        get web server credentials ()
        connect to wifi()
        if(no connect)
        {
                Start wifi AP mode to get connected()
        }
        get switch status()
        Start loop()
}
void loop() {
        if(connected ){
                check web server status()
                check mobile app status()
                check manual switch status()
        }
        update relay status()
        check status of switch()
        delay()
        continue loop()
}
```

A general flow of code is mention above. Origenal code contain different checks and data storage which are not mention in sudo code.

4) Web sevice format and data storage format.

GET and POST request and response from switch and mobile app to web server.

Check request from mobile app to web server

Request URL: https://smart-fox.info?key=wasdhlkhasdd&switchid=14920

Request Method: GET

Status Code: 200

Remote Address: 128.30.52.100:443

Referrer Policy: origin

Responce = 14920:true

Update request from mobile app to web server

Request URL: https://smart-fox.info

Request Method: POST

Status Code: 200

Remote Address: 128.30.52.100:443

Referrer Policy: origin

Date: key=wasdhlkhasdd & id=14920 & stutus=true

Update request from mobile app to smart switch.

Request URL: https://192.168.43.121:5000 // local ip of smart switch

Request Method: POST

Status Code: 200

Remote Address: 128.30.52.100:443

Referrer Policy: origin

Date: key=wasdhlkhasdd & id=14920 & stutus=true

Update request from switch to web server

Request URL: https://smart-fox.info

Request Method: POST

Status Code: 200

Remote Address: 128.30.52.100:443

Referrer Policy: origin

Date: key=wasdhlkhasdd & id=14920 & stutus=true & deviceid = A400006

check request from switch to web server for all switch

Request URL: https://smart-fox.info

Request Method: POST

Status Code: 200

Remote Address: 128.30.52.100:443

Referrer Policy: origin

Date: key=wasdhlkhasdd & deviceid = A400006

Responce = 1:true, 2:false, 3:true, 4: false

5) Mobile application flow and development.

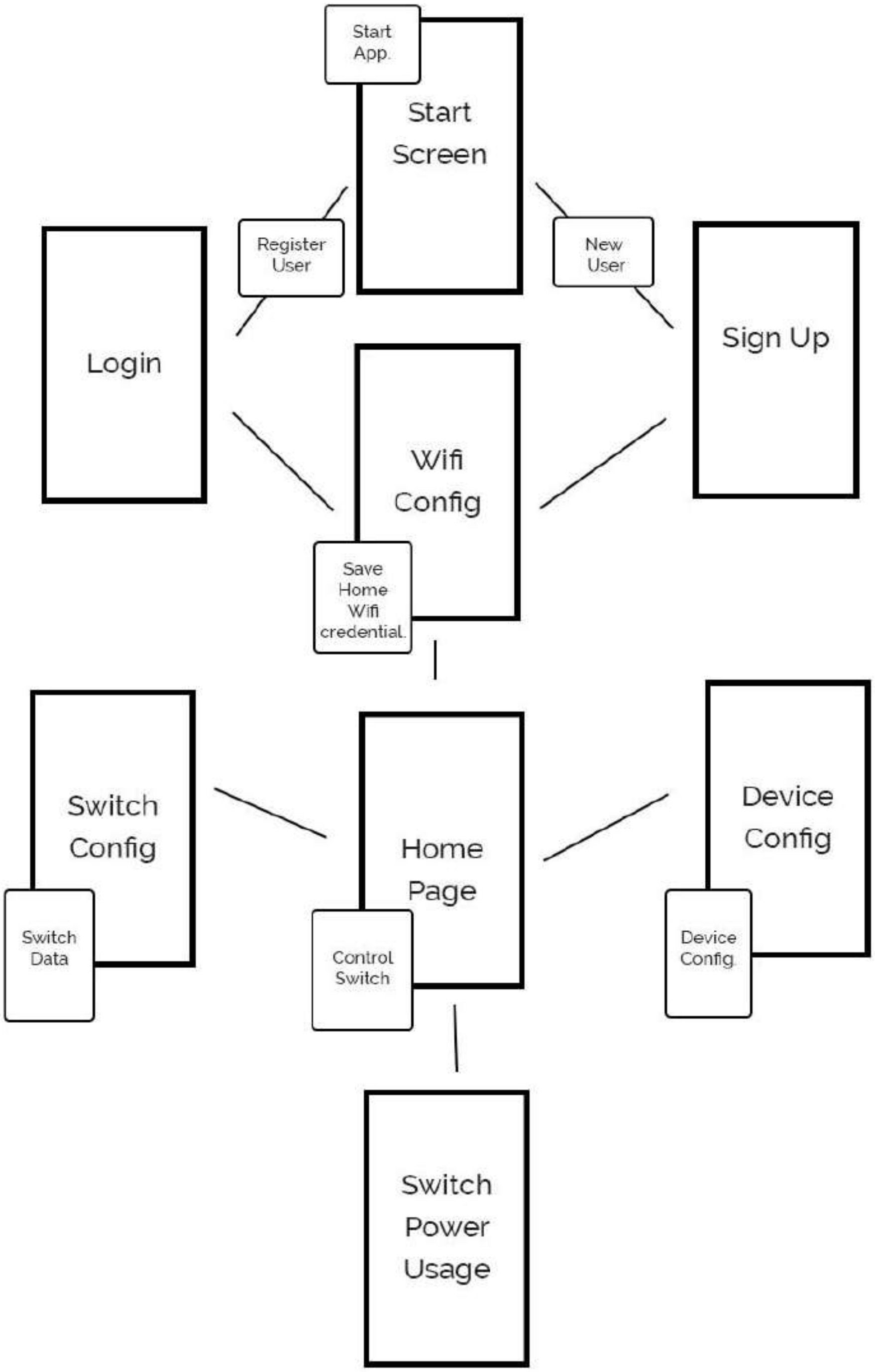


Fig 4.8 Flow chart of mobile application

Start Screen - Inital load all the data from room SQLite database from mobile and check varify login credentials and sync data with web server.

Login Screen - Check login credentials and redirect to Wi-Fi configuration screen.

Signup Screen - Submit new user login detail with email opt varification.

Wifi Screen - Save Wi-Fi creadentials localy, which help to device cofiguration to connect device to local Wi-Fi network.

Home Screen - where user can control all configure switch. When connected to network.

Device Config. **-** Configure new switch device to app and connect switch to local Wi-Fi network, when switch is in Wi-Fi AP mode.

Switch Config. - When new switch device is configure to app it automatically add all device switch to app and configure at default mode. Which can edit my user as per need.

Switch Power usage - Shows to detail utilization of power for each switch and device in Kilo-watts hours over a time of year.

4.2.1 IP-based Surveillance Camera development.

1) Hardware circuit development

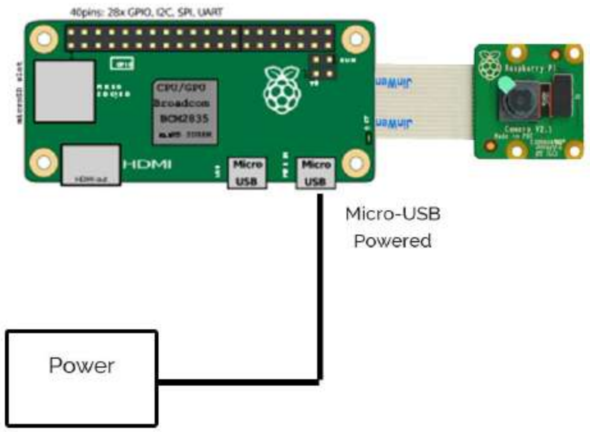


Fig 4.9 Camara Setup

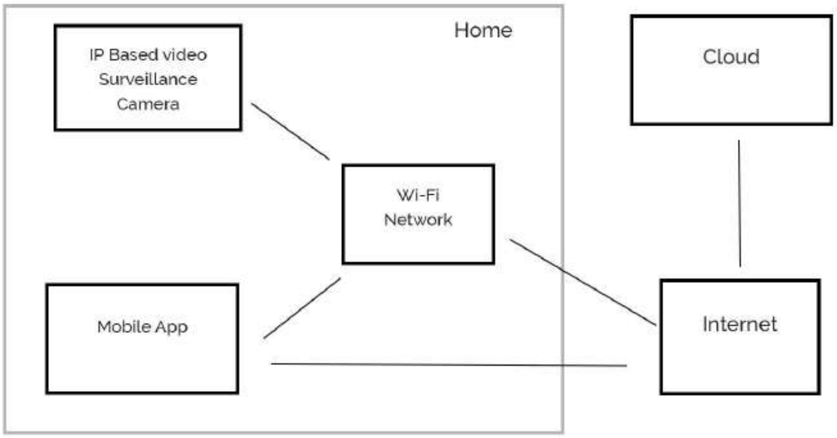


Fig 4.10 Surveillance flow diagram

2) Different test case parameters.

- Image processing - Motion Detection, video quality enhacement.
- Video storage mangement in-build memory storage.
- Live streaming video on mobile app.
- Cloud communication.

3) Sudo program.

```
import opencv library
void start()
{
    initial image capturing()
    connect to wifi()
    start loop()
}
void  loop()
{
    if(wifi connected)
    {
        connect to cloud()
        start live steaming web service()
    }
    if(motion detected())
    {
        store image frame()
    }
}
```

4) Web sevice format and data storage format.

Video data flow processing

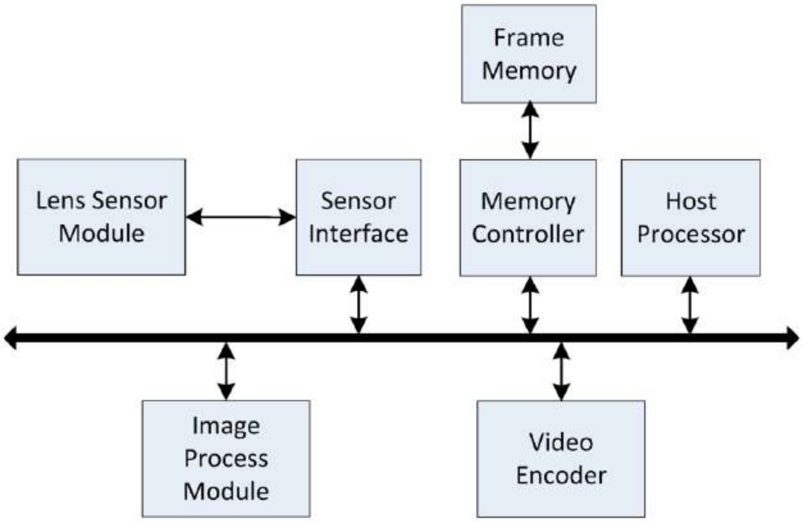


Fig 4.11 block processes on video

Selection of right stroage formate. [4]

UDP – User Datagram Protocol is a transport protocol that can deliver “datagrams” (data with some networking metadata attached to it) from one computer to another. The data in a datagram might be just a small chunk of the full data object being transferred between two computers, so each datagram has a sequence number that aids in properly reconstructing the original data object. Unlike TCP, the UDP does not require a connection to be established and kept alive between two computers. One computer informs another computer of its IP address and the port on which it is expecting to receive datagrams, and as long as a socket is listening on that port the datagrams can keep flowing in. Another difference between UDP and TCP is that the former does not make any guarantees that datagrams will arrive in the correct order (or arrive at all). It is the application developer’s job to deal with reassembling the data and accommodating any missing datagrams. UDP is commonly used for applications that do not benefit from the strong delivery guarantees of TCP, such as video streaming where dropping an occasional video frame or two is not a noticeable problem.

RTSP – Real-Time Streaming Protocol is like HTTP but it is built specifically for setting up, configuring, and tearing down streaming media sessions, such as a video chat session or a movie being streamed to a computer from streaming media server. RTSP is not involved with the actual video streaming, but just managing the streaming session. It has a set of directives like DESCRIBE, which asks the streaming media source to list the available media types and their capabilities, and SETUP, which is how the client app tells the server which media type it wants to consume and gets a session ID back in the response.

RTP – Real-time Transport Protocol is the low-level protocol used to break a video into packets that can be streamed between computers. Those packets of video data are often put into UDP datagrams. To get a feel for the kind of processing involved with RTP video streams, check out this high-level overview of the process on StackOverflow.

H.264 and MJPEG – These are commonly used compression techniques for audio/video files. Most normal humans do not ever need to know or care about how such things work or can be decoded, since companies like Apple build support for them into their products. iOS uses hardware acceleration to decode H.264 video (by processing it on the GPU instead of the CPU). I'm not sure if they hardware accelerate MPEG4 but I assume they do. For a taste of what's involved with decoding H.264 formatted for RTP, check out RFC 6184. The main advantage of MPEG is high quality video at fairly low data rates , In properly equipped pcs , this is an excellent solution , We can size/position the video anywhere on the screen in both AVI and MPEG format .

For this project selecting MJPEG will be advenatages protocal to live stream and store video.

4.2 Prototype

1) Smart Switch

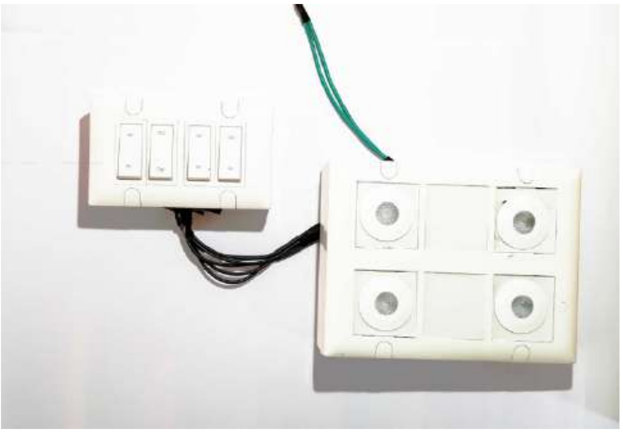
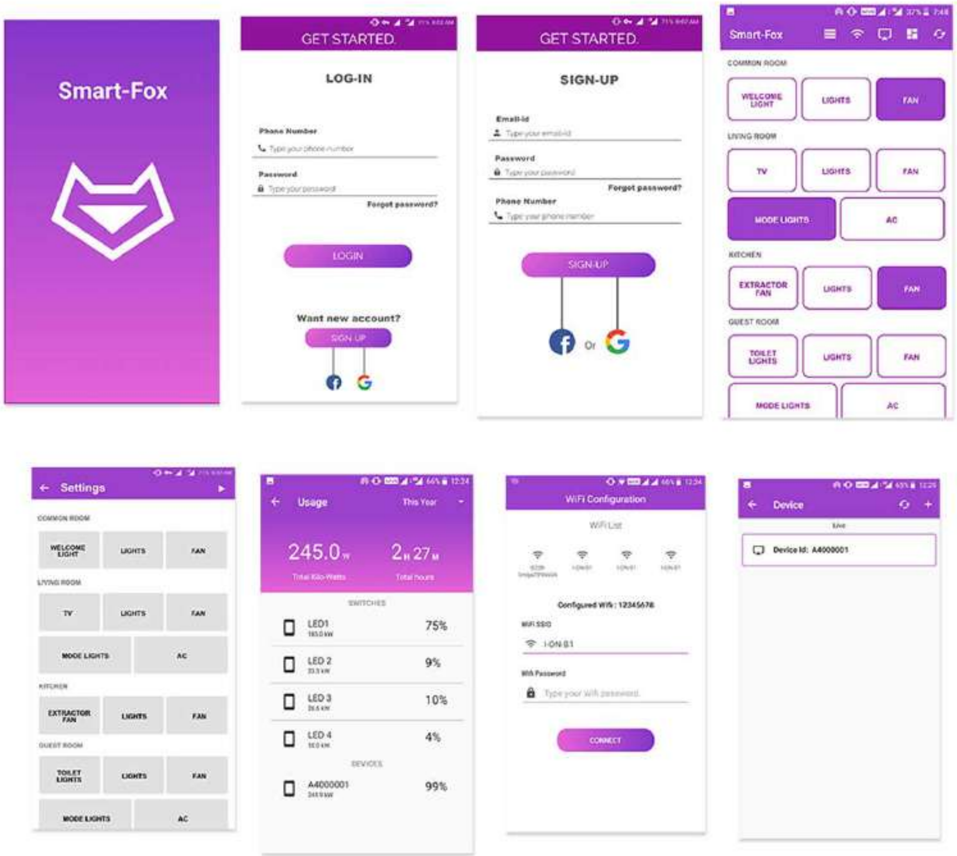


Fig 4.12 Smart switch prototype



245.0 W Total Kilo-Watts
2 H 27 M Total hours

SWITCHES

LED1	75%
LED 2	9%
LED 3	10%
LED 4	4%

DEVICES

A4000001	99%

Configured Wifi : 12345678

Wifi SSID: I-ON B1

Device Id: A4000001

Fig 4.13 Smart switch mobile application

2) IP-based surveillance camera

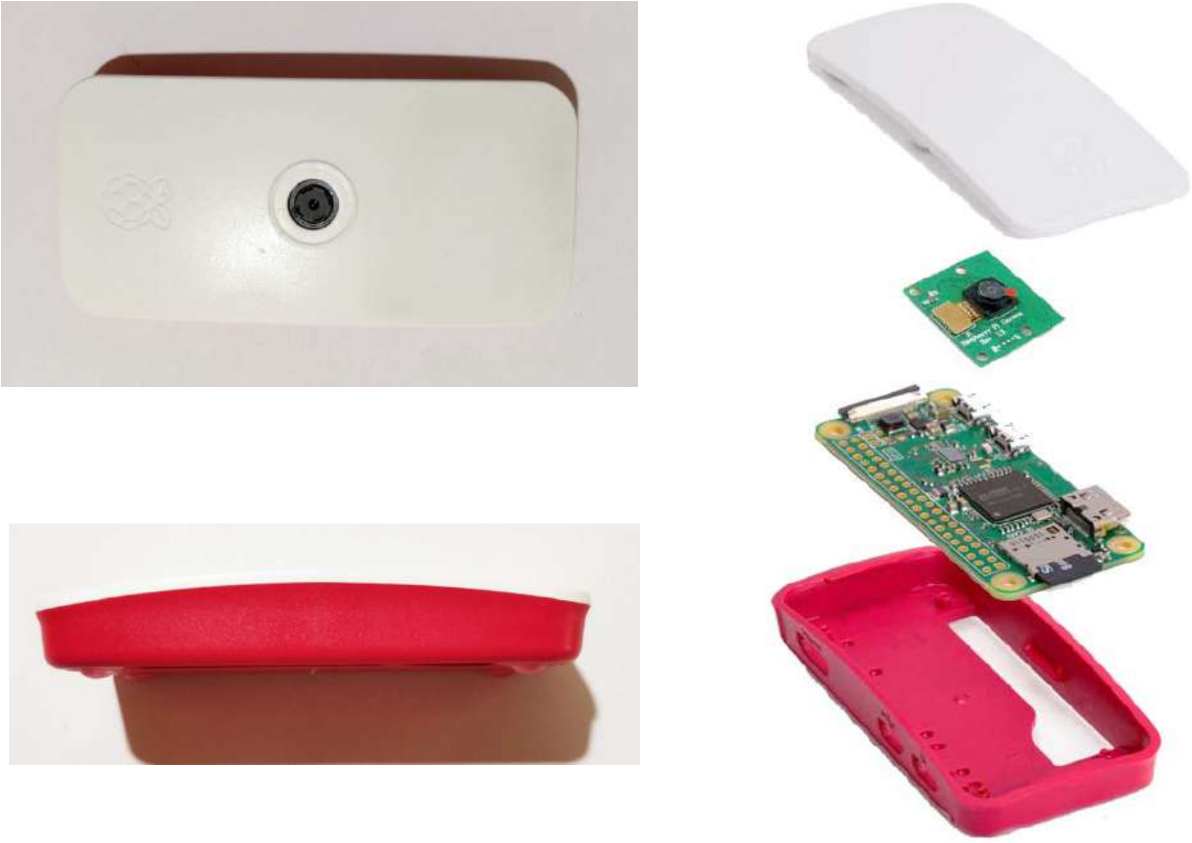


Fig 4.14 IP-based Surveillance Camera Prototype

Last Chapter

CONCLUSION & FUTURE SCOPE

6.1 Conclusions

The prime objective of our project is to use the Smartphone to control the home appliances effectively. The switch mode are used to control the home appliances. The video feedback is received in the android app which streams the video of IP- Camera. This project is based on the Raspberry pi, Android platform Java, NodeMCU and Python. These platforms are Free Open Source Software. So the overall implementation cost is low and can be easily configured. User can easily interact with the android phone/tablet. The user can send commands via the mobile application. The data are being analyzed by the application and are sent over a network. The Raspberry pi and NodeMCU acts as a hardware, analyses the data and activates the GPIO (General Purpose Input Output) Pins. The GPIO Pins are connected to the relays switch which activated the required home appliances. In this way, automation process is carried out.

Both device are cost low extremely low.

Device	**Price**
Smart-Switch	550 Rs (4 switch)
IP-Based Surveillance	3,000 Rs (per unit with 24 hours video storage)

6.2 Future Scope of Work

This system is designed to assist and provide support in order to fulfill the needs of elderly and disabled in home. Household appliances can be easily controlled via a Mobile/Tablet. Status of light, fan and other electrical appliances can be known. With the help of IP camera, video of rooms or certain area of a house can be recorded. This helps to provide security.

REFERENCES

Conference Papers

[1] , “Home Automation”, Energy Efficient Smart Home Automation System volume no 1, publication year 2015, Total pages 11.

Web

[1] About ESP8266, www.espressif.com

[2] Raspberry Pi Zero W, www.raspberrypi.org

[3] Camera Sensor, www.sony-semicon.co.jp

[4] Software development cycle, www.upedu.org

[5], NodeMCU, www.nodemcu.com

[6], Raspberry Pi Zero W setup, www.developer.ibm.com

[7], Implementation of OpenCV using python, docs.opencv.org

[8], Motion detection in video using python, www.technicdynamic.com

[9], Video quality enhancement using OpenCV, www.opencv-python-tutroals.readthedocs.io/

[10], Video storage in H.264, www.learnopencv.com